

EV3 Sensor API for NXC4EV3

The new sensor API is composed of these three functions:

int SetSensorEV3Type(port, type)

- set sensor type (color, ultrasonic, etc.)

int SetSensorEV3Mode(port, mode)

- set sensor mode (angular momentum, angular position, etc.)

int SensorValueSlotted(port, slot)

- return sensor value on specified slot and port

Sensors can return multiple values at once. This is what slots are for. Each sensor mode can return multiple values, one per slot. They are usually named according to the value they contain. If the mode has only one slot, you can use also this function:

int SensorValue(port)

Tested way of specifying ports it following:

- *IN_1*
- *IN_2*
- *IN_3*
- *IN_4*

Sensor summary:

- **Ultrasonic sensor** = *EV3_Sonic*

modes:

- *Sonic_Continuous_Mm*
 - Returns the distance between an obstacle and the sensor in millimeters. If the measurement is out of range, *SONIC_NONE_MM* is returned.
 - only slot *Sonic_Only*
- *Sonic_Continuous_In*
 - Returns the distance between an obstacle and the sensor in inches. If the measurement is out of range, *SONIC_NONE_IN* is returned.
 - only slot *Sonic_Only*
- *Sonic_Single_Mm*
 - Makes one distance measurement during the mode switch and returns the measured value in millimeters. Out-of-range returns *SONIC_NONE_MM*.
 - only slot *Sonic_Only*
- *Sonic_Single_In*
 - Makes one distance measurement during the mode switch and returns the measured value in inches. Out-of-range returns *SONIC_NONE_IN*.
 - only slot *Sonic_Only*
- *Sonic_Listen*
 - Detects whether there is another ultrasonic sensor running nearby. Zero is returned if not.
 - only slot *Sonic_Only*

□ **Touch sensor** = *EV3_Touch*

modes:

- *Touch_Touch*
 - Detects whether the button is pressed (1 or 0)
 - only slot *Touch_Only*

□ **Color sensor** = *EV3_Color*

modes:

- *Color_Reflect*
 - Returns amount of reflected red light. Values are in range from 0 to 100.
 - only slot *Color_Only*
- *Color_ReflectRaw*
 - Undocumented version of *Color_Reflect* returning sensor-internal values.
 - slots:
 - *Color_RrawReflect*
 - *Color_RrawBkgnd*
- *Color_Ambient*
 - Returns amount of sunlight shining on the sensor. Values are in range from 0 to 100.
 - only slot *Color_Only*
- *Color_ColorId*
 - Identifies the color of the object that is in front of the color sensor.
 - only slot *Color_Only*
 - returned values:
 - *CID_NONE*
 - *CID_BLACK*
 - *CID_BLUE*
 - *CID_GREEN*
 - *CID_YELLOW*
 - *CID_RED*
 - *CID_WHITE*
 - *CID_BROWN*
- *Color_ColorRaw*
 - Returns raw internal RGB measurements from the sensor.
 - slots:
 - *Color_CrawRed* – amount of reflected red light
 - *Color_CrawGreen* – amount of reflected green light
 - *Color_CrawBlue* – amount of reflected blue light

□ **Gyro sensor** = *EV3_Gyro*

modes:

- *Gyro_Angle*
 - Returns angular position difference from the last sensor mode change.
 - slot only *Gyro_Only*
- *Gyro_Rate*
 - Returns current angular velocity.
 - slot only *Gyro_Only*
- *Gyro_AngleRate*
 - Returns both the angular position and velocity.
 - slots:
 - *Gyro_AngleSlot* – angular position difference
 - *Gyro_RateSlot* – angular velocity
- *Gyro_Fas*
 - Undocumented sensor-internal value.

- only slot *Gyro_Only*
- Infrared sensor = EV3_Infra**
modes:
 - *Infra_Proximity*
 - Returns the proximity to the remote control in percent.
 - only slot *Infra_Only*
 - *Infra_Seek*
 - Seeking of up to four remote controls.
 - *Head* → heading from the sensor to the RC, range: -25 is far left, +25 is far right.
 - *Dist* → RC proximity; INFRA_NONE if the RC was not detected.
 - slots:
 - Infra_Seek_HeadA*
 - Infra_Seek_DistA*
 - Infra_Seek_HeadB*
 - Infra_Seek_DistB*
 - Infra_Seek_HeadC*
 - Infra_Seek_DistC*
 - Infra_Seek_HeadD*
 - Infra_Seek_DistD*
 - *Infra_Remote*
 - Reads pressed keys on up to four remote controls.
 - slots:
 - Infra_RemoteA*
 - Infra_RemoteB*
 - Infra_RemoteC*
 - Infra_RemoteD*
 - returned values: (R is red axis, B is blue axis; the button is pressed when mentioned)
 - IR_STATE_NONE*
 - IR_STATE_RUP*
 - IR_STATE_RDOWN*
 - IR_STATE_BUP*
 - IR_STATE_BDOWN*
 - IR_STATE_RUP_BUP*
 - IR_STATE_RUP_BDOWN*
 - IR_STATE_RDOWN_BUP*
 - IR_STATE_RDOWN_BDOWN*
 - IR_STATE_BEACON_ON*
 - IR_STATE_RUP_RDOWN*
 - IR_STATE_BUP_BDOWN*



For a quick illustration, we can use the following demo program. At the beginning of the main function the sensors are configured. Then, the sensor values are being read and then written on the screen in the main loop.

```
#define L(x) (((x)-1)*10)

void output(int line, string what, int value) {
    TextOut(0, L(line), what);
    NumOut(60, L(line), value);
}

task main(){
    SetSensorEV3Type(IN_4, EV3_Touch);
    SetSensorEV3Mode(IN_4, Touch_Touch);
    SetSensorEV3Type(IN_3, EV3_Color);
    SetSensorEV3Mode(IN_3, Color_Reflect);

    while (true) {
        int pressed = SensorValueSlotted(IN_4, Touch_Only);
        int reflected = SensorValueSlotted(IN_3, Color_Only);

        output(1, "PRESS", pressed);
        output(2, "LIGHT", reflected);

        Wait(10);
    }
}
```